NAME

kstat – kernel statistics structure

SYNOPSIS

#include <sys/types.h>
#include <sys/kstat.h>
#include <sys/ddi.h>
#include <sys/sunddi.h>

INTERFACE LEVEL

Solaris DDI specific (Solaris DDI)

DESCRIPTION

Each kernel statistic (kstat) exported by device drivers consists of a header section and a data section. The **kstat** structure is the header portion of the statistic.

A driver receives a pointer to a **kstat** structure from a successful call to **kstat_create**(9F). Drivers should never allocate a **kstat** structure in any other manner.

After allocation, the driver should perform any further initialization needed before calling **kstat_install**(9F) to actually export the kstat.

STRUCTURE MEMBERS

```
void      *ks_data;                          /* kstat type-specific data */
ulong_t   ks_ndata;                          /* # of type-specific data records */
ulong_t   ks_data_size;                      /* total size of kstat data section */
int       (*ks_update)(struct kstat *, int);
void      *ks_private;                       /* arbitrary provider-private data */
void      *ks_lock;                          /* protects this kstat's data */
```

The members of the **kstat** structure available to examine or set by a driver are as follows:

ks_data points to the data portion of the kstat. Either allocated by **kstat_create**(9F) for the drivers use, or by the driver if it is using virtual kstats.

ks_ndata is the number of data records in this kstat. Set by the **ks_update**(9E) routine.

ks_data_size is the amount of data pointed to by **ks_data**. Set by the **ks_update**(9E) routine.

ks_update is a pointer to a routine which dynamically updates kstats. This is useful for drivers where the underlying device keeps cheap hardware stats, but extraction is expensive. Instead of constantly keeping the kstat data section up to date, the driver can supply a **ks_update**(9E) function which updates the kstat's data section on demand. To take advantage of this feature, set the **ks_update** field before calling **kstat_install**(9F).

ks_private is a private field for the driver's use. Often used in **ks_update**(9E).

ks_lock is a pointer to a mutex that protects this kstat. kstat data sections are optionally protected by the per-kstat **ks_lock**. If **ks_lock** is non-**NULL**, kstat clients (such as **/dev/kstat**) will acquire this lock for all of their operations on that kstat. It is up to the kstat provider to decide whether guaranteeing consistent data to kstat clients is sufficiently important to justify the locking cost. Note, however, that most statistic updates already occur under one of the provider's mutexes, so if the provider sets **ks_lock** to point to that mutex, then kstat data locking is free. **ks_lock** is really of type **(kmutex_t *);** it is declared as **(void *)** in the kstat header so that users don't have to be exposed to all of the kernel's lock-related data structures.

SEE ALSO

kstat_create(9F)

Writing Device Drivers